



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/600,906

06/20/2003

Jerome M. Verbeke

5681-65900

9206

58467

7590

11/28/2007

MHKKG/SUN

P.O. BOX 398

AUSTIN, TX 78767

EXAMINER

CAO, DIEM K

ART UNIT

PAPER NUMBER

2194

MAIL DATE

DELIVERY MODE

11/28/2007

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/600,906

Applicant(s)

VERBEKE ET AL.

Examiner

Diem K. Cao

Art Unit

2194

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 12 September 2007.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-9, 11-27, 29-43, 45-60 and 62-68 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-9, 11-27, 29-43, 45-60 and 62-68 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

WILLIAM THOMSON
SUPERVISORY PATENT EXAMINER

Attachment(s)

- | | |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. Claims 1-9, 11-27, 29-43, 45-60 and 62-68 are pending. Applicant has amended claims 1, 11, 18, 29, 34, 35, 45, 52 and 62 and cancelled claims 10, 28, 44 and 61.

Continued Examination Under 37 CFR 1.114

2. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 9/12/2007 has been entered.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

4. **Claims 1-8, 11-13, 16, 34-42, 45-47, 50, 52-59, 62-64 and 67 are rejected under 35 U.S.C. 103(a) as being unpatentable over Monday et al. (U.S. 6,263,377 B1) in view of Venners (Inside the Java Virtual Machine).**

As to claim 1, Monday teaches a system, comprising:

- a processor (a central processing unit 101; col. 2, lines 22-23); and

- a memory comprising program instructions, wherein the program instructions are executable by the processor to implement (memory, distributed application manager 132; col. 2, lines 24-25, 34-36):

- a default class loader (CLASSLOADER; col. 3, line 44) configured to:

- load classes for code on the system from one or more local locations indicated by a class path of the default class loader (checks the environment variable CLASSPATH for ... to the requesting application; col. 3, lines 38-43); and

- determine that a class needed to execute the code on the system is not stored in the one or more locations indicated by the class path (checks the environment variable CLASSPATH ... selected file; col. 3, lines 38-40 and If x.class is not loaded; col. 3, line 43);

- a remote class loader mechanism configured to (a REMOTECLASSLOADER; col. 3, line 45):

- obtained the class from a remote system via a network (a REMOTECLASSLOADER checks ... if the class is found; col. 3, lines 45-54); and

- store the class in a location indicated by the class path of the default class loader on the system (write it to the first CLASSPATH directory, thus building the class locally that the network does not have to be consulted on the next run; col. 3, lines 54-56);

- wherein the remote class loader mechanism is configured perform the obtain and store separate from and transparent to the default class loader (inherent from the remote class loader check the remoteclasspath, obtain the class and store it in the directory without consulting from the class loader; col. 3, lines 44-56), and

- wherein the default class loader is configured to load the class from the location

indicated by the class path (inherent from "checks the environment variable CLASSPATH for ... to the requesting application"; col. 3, lines 38-43, since only the CLASSLOADER can load from the CLASSPATH).

Monday does not explicitly teach a virtual machine, the default class loader for the virtual machine, and code executable within the virtual machine, generate an indication that the class is not loaded, and detect the indication that the class is not loaded. However, Monday teaches the application is written in Java programming language (col. 2, lines 61-62). Verners teaches Java application runs inside a Java virtual machine (Chapter 4, page 1, section What is a Java Virtual Machine), each Java virtual machine has a class loader subsystem for loading codes to execute inside the virtual machine (chapter 5, page 3), generate an indication that the class is not loaded (ClassNotFoundException; chapter 5, pages 11-12, section "User-Defined Class Loaders"), and detect the indication that the class is not loaded separate and transparent from the default class loader (ClassNotFoundException; chapter 5, pages 11-12, section "User-Defined Class Loaders" and Chapter 3 "Security", page 5, second paragraph).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teaching of Verners to the system of Monday because Verners teaches in details how a Java application load and utilize class that are needed at runtime.

As to claim 2, Monday teaches the default class loader is configured to locate the class stored in the location indicated by the class path, and load the class from the location for access by the code (checks the environment variable CLASSPATH for ... to the requesting application; col. 3, lines 38-43).

As to claim 3, Venners teaches the location is a default directory for storing remote classes (The class loader ... each class loader; chapter 3, page 4 and page 6).

As to claim 4, Venners teaches the location is a user-specified directory for storing remote classes (user-defined directory path; chapter 5, page 11, section “The Bootstrap Class Loader”).

As to claim 5, Venners teaches the indication is an exception generated by the code and indicating that the class is not stored in the one or more locations indicated by the class path (ClassNotFoundException; chapter 5, pages 11-12, section “User-Defined Class Loaders”).

As to claim 6, Monday teaches to obtain the class from a remote system, the remote class loader mechanism is further configured to send a message requesting the class to one or more remote systems, wherein the message comprises information about the class for identifying a class file on the remote system that comprises the requested class (a REMOTECLASSLOADER checks ... if the class is found; col. 3, lines 45-54).

As to claim 7, Monday teaches to obtain the class from a remote system, the remote class loader mechanism is further configured to send a message requesting the class to the remote system, and receive the class from the remote system in one or more messages in response to the message (a REMOTECLASSLOADER checks ... if the class is found; col. 3, lines 45-54).

As to claim 8, Monday teaches to obtain the class from a remote system, the remote class loader mechanism is further configured to broadcast a message requesting the class to one or more remote systems including the remote system on the network, and receive the class from the remote system in one or more messages in response to the broadcast message (a REMOTECLASSLOADER checks ... if the class is found; col. 3, lines 45-54).

As to claim 11, Venners teaches the virtual machine is a Java Virtual Machine (Java Virtual Machine; Chapter 5 "The Java Virtual Machine", page 1).

As to claim 12, Monday as modified by Venners teaches the code is in a bytecode computer language (Java; col. 2, lines 61-62).

As to claim 13, Monday teaches the code is Java code (Java; col. 2, lines 61-62).

As to claim 16, Monday teaches the code a code fragment of an application configured for execution on the system, and wherein the remote system is a node in a distributed computing framework that comprises the application and is configured to provide computer-executable code fragments of the application to two or more other systems to run the code fragments in parallel to execute the application (abstract).

As to claim 34, Monday teaches

- a default class loader is configured to load classes for code on the system from one or more local locations indicated by a class path of the default class loader (CLASSLOADER, checks the environment variable CLASSPATH for ... to the requesting application; col. 3, lines 38-44);

- means for determining that a class needed to execute the code on the system is not stored in the one or more locations indicated by the class path (checks the environment variable CLASSPATH ... selected file; col. 3, lines 38-40 and If x.class is not loaded; col. 3, line 43);

- means for obtaining the class from a remote system via a network (a REMOTECLASSLOADER checks ... if the class is found; col. 3, lines 45-54); and

- means for storing the class in a location on the system indicated by the class path of the default class loader means (write it to the first CLASSPATH directory, thus building the class locally that the network does not have to be consulted on the next run; col. 3, lines 54-56);

- wherein the means for determining, the means for obtaining, the means for storing are configured to operate separate from and transparent to the default class loader (inherent from the remote class loader check the remoteclasspath, obtain the class and store it in the directory without consulting from the class loader; col. 3, lines 44-56), and

- wherein the default class loader is configured to load the class from the location indicated by the class path (inherent from "checks the environment variable CLASSPATH for ... to the requesting application"; col. 3, lines 38-43, since only the CLASSLOADER can load from the CLASSPATH).

Monday does not explicitly teach the default class loader for the virtual machine, and

loading classes for codes executable within the virtual machine. However, Monday teaches the application is written in Java programming language (col. 2, lines 61-62). Verners teaches Java application runs inside a Java virtual machine (Chapter 4, page 1, section What is a Java Virtual Machine), each Java virtual machine has a class loader subsystem for loading codes to execute inside the virtual machine (chapter 5, page 3). Verners further teaches generate an indication that the class is not loaded (ClassNotFoundException; chapter 5, pages 11-12, section "User-Defined Class Loaders"), and detect the indication that the class is not loaded separate and transparent from the default class loader (ClassNotFoundException; chapter 5, pages 11-12, section "User-Defined Class Loaders" and Chapter 3 "Security", page 5, second paragraph).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teaching of Verners to the system of Monday because Verners teaches in details how a Java application load and utilize class that are needed at runtime.

As to claim 35, it is the same as the system claim of claim 1 except it is a method claim, and is rejected under the same ground of rejection.

As to claims 36-42, 45-47 and 50, see rejections of claims 2-8, 11-13 and 16, respectively.

As to claim 52, it is the same as the system claim of claim 1 except this is a computer product claim, and is rejected under the same ground of rejection.

As to claims 53-59, 62-64 and 67, see rejections of claims 2-8, 11-13 and 16 above.

5. Claims 9, 14, 15, 17-27, 29-33, 43, 48, 49, 51, 60, 65, 66 and 68 are rejected under 35 U.S.C. 103(a) as being unpatentable over Monday et al. (U.S. 6,263,377 B1) in view of Venners (Inside the Java Virtual Machine) further in view of Babaoglu et al. (Anthill: A Framework for the Development of Agent-Based Peer-to-Peer Systems).

As to claim 9, Monday does not teach the one or more remote systems and the system are member peers of a peer group in a peer-to-peer network environment. However, Babaoglu teaches peer-to-peer application can be implemented in Java (page 7, section 4). It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teaching of Babaoglu to the system of Monday because it presents a framework supporting anew approach for building P2P application in which resource can be sharing by direct exchange between peer nodes.

As to claim 14, see rejection of claim 9 above.

As to claim 15, Monday does not teach the system and the remote system are configured to participate as peer nodes in a peer-to-peer platform protocols for enabling the peer nodes to discover each other, communicate with each other, and cooperate with each other to form peer groups in the peer-to-peer environment. However, Babaoglu teaches peer-to-peer application can be implemented in Java (page 7, section 4).

As to claim 17, Monday teaches the system and the remote system are configured to participate in a distributed computing system on the network (abstract). Monday does not teach submitting computational tasks in a distributed heterogeneous networked environment that utilizes peer groups to decentralize task dispatching and post-processing functions and enables a plurality of jobs to be managed and run simultaneously. However, Babaoglu teaches peer-to-peer application can be implemented in Java (page 7, section 4). It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teaching of Babaoglu to the system of Monday because it presents a framework supporting new approach for building P2P application in which resource can be sharing by direct exchange between peer nodes.

As to claim 18, see rejections of claim 1 and 17 above.

As to claims 19-28 and 29-33, see rejections of claims 2-9 and 11-17 above.

As to claim 43, see rejection of claim 9 above.

As to claims 48-49, see rejections of claims 14-15 above.

As to claim 51, see rejection of claim 17 above.

As to claim 60, see rejection of claim 9 above.

As to claims 65-66, see rejection of claims 14-15 above.

As to claim 68, see rejection of claim 17 above.

Response to Arguments

6. Applicant's arguments filed 9/12/2007 have been fully considered but they are not persuasive.

In the remarks, Applicant argued in substance that (1) Monday does not teach nor suggest “a default class loader means for a virtual machine, wherein the default class loader means is configured to load classes for code executable within the virtual machine on the system from one or more local locations indicated by a class path of the default class loader means” since Monday discloses a distributed application manager checks the variable CLASSPATH for a set of directories to browser for a selected class file (pages 18-19), (2) Monday does not teach the limitations of claim 34 (page 20, second paragraph), (3) Claim 34 recites means for implementing a mechanism to remotely load classes needed to run, an application in a distributed computing environment through the default class loader of the virtual machine, not a custom class loader, which is in contrast with Monday teaches (page 21-22), (4) no motivation to combine the teaching of Venners and Monday, (5) no motivation to apply the teaching of Babaoglu to the system of Monday and Venners.

Examiner respectfully disagrees with the Applicant's arguments:

- As to the point (1), the limitation is newly added and is now taught by Monday and Venners. Monday teaches the applications and the distributed application manager are written in Java language, and Venners teaches the Java codes are executed inside a Java virtual machine, and the Java virtual machine includes default class loader and custom class loaders. Furthermore, Monday teaches the remote class loader loads the class from a remote server and store it in the location of the classpath of the default class loader, thus, when the application is executed, the default class loader would have loaded from the classpath location. Thus, Monday and Venners teach the limitations.
- As to the point (2), Applicant fails to show how the cited passages do not teach the claim limitations, therefore, the arguments are not persuasive.
- As to the point (3), Monday teaches the remote class loader loads the class from a remote server and store it in the location of the classpath of the default class loader, thus, when the application is executed, the default class loader would have loaded from the classpath location. Thus, Monday and Venners teach the limitations.
- As to the points (4) and (5), In response to applicant's argument that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*,

837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988) and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, Monday and Venners teaches default class loader, custom/user-defined class loader for the Java language environment, thus, one of ordinary skill in the art would apply the teaching of Venners to the system of Monday for more teaching into the Java virtual machine which include class loader for more information. Babaoglu teaches Java application can be implemented as peer-to-peer application, thus, they are all in the same field of technology.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Diem K. Cao whose telephone number is (571) 272-3760. The examiner can normally be reached on Monday - Friday, 8:30AM - 4:30PM.


If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William Thomson can be reached on (571) 272-3718. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Application/Control Number:
10/600,906
Art Unit: 2194

Page 14

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

DC
November 19, 2007


WILLIAM THOMSON
SUPERVISORY PATENT EXAMINER